

Software Reuse Within the Earth Science Community

James J. Marshall, Stephen W. Olding, Robert E. Wolfe

NASA Goddard Space Flight Center
Greenbelt, MD, USA

james.marshall@gsfc.nasa.gov, solding@everware.com,
robert.e.wolfe@nasa.gov

Victor E. Delnore

NASA Langley Research Center
Hampton, VA, USA

v.e.delnore@nasa.gov

Abstract—Scientific missions in the Earth sciences frequently require cost-effective, highly reliable, and easy-to-use software, which can be a challenge for software developers to provide. The NASA Earth science community spends a significant amount of resources developing software components and other software development artifacts that may also be of value if reused in other projects requiring similar functionality. In general, software reuse is often defined as utilizing existing software artifacts. Software reuse can improve productivity and quality while decreasing the cost of software development, as documented by case studies in the literature. Since large software systems are often the results of the integration of many smaller and sometimes reusable components, ensuring reusability of such software components becomes a necessity. Indeed, designing software components with reusability as a requirement can increase the software reuse potential within a community such as the NASA Earth science community.

The NASA Earth Science Data Systems (ESDS) Software Reuse Working Group is chartered to oversee the development of a process that will maximize the reuse potential of existing software components while recommending strategies for maximizing the reusability potential of yet-to-be-designed components. As part of this work, two surveys of the Earth science community were conducted. The first was performed in 2004 and distributed among government employees and contractors. A follow-up survey was performed in 2005 and distributed among a wider community, to include members of industry and academia. The surveys were designed to collect information on subjects such as the current software reuse practices of Earth science software developers, why they choose to reuse software, and what perceived barriers prevent them from reusing software.

In this paper, we compare the results of these surveys and discuss the findings, including the difference between the components desired for reuse and those made available for reuse. The results are very similar, with the second, larger survey confirming the basic results of the first, smaller survey. The results suggest that reuse of Earth science software can drive down the cost and time of system development, increase flexibility and responsiveness of these systems to new technologies and requirements, and increase effective and accountable community participation.

Software reuse; Earth science; SEEDS; NASA

I. INTRODUCTION

Software reuse is the reapplication of a variety of kinds of knowledge about one system to another system in order to reduce the effort of developing and maintaining that system. In

principle, many different artifacts produced during the software development life cycle can be reused. Some typical examples of reusable artifacts include source code, analysis and design specification, plans, data, documentation, expertise and experience, and any information used to create software and software documentation. While all of these items are useful, the most often reused artifacts are software components.

Software reuse is used in order to realize a number of potential benefits including reducing cost, saving time, and increasing reliability [1, 2]. Productivity and quality improvements are also typical motivations for reuse [3]. Productivity is often measured in terms of cost and labor, and reuse has the potential to decrease both, thereby increasing productivity. Increased productivity can also be used to reduce the time needed to start using the software, an important factor in the competitive research environment. Reusing software can also improve the reliability and quality of new products because the currently existing software components have already been tested and confirmed to perform according to their designs.

The NASA Earth Science Data System (ESDS) Software Reuse Working Group is chartered to oversee the development of a process that will maximize the reuse potential of existing software components while recommending strategies for maximizing the reusability potential of yet-to-be-designed components. As part of this work, we conducted two surveys of members of the Earth science community in order to get a measure of their reuse practices. Here, we describe these surveys, compare their results, and discuss the findings.

II. SURVEY DESCRIPTION

We conducted two surveys, the first in 2004 and the second in 2005. They were identical with the exception of one question that was added to the 2005 survey. The survey questions were grouped into four major categories – background information, recent reuse experiences, reusability / developing for reuse, and community needs. The background section included questions on the respondent's role in software development, organization, operating systems used or planned for use, and programming languages used. The questions on recent reuse experience asked if respondents did or did not reuse artifacts from outside their project or group within the last five years, then followed up with questions including why they did or did not reuse components, what they reused, the factors influencing their decision to reuse, and where they

This work was sponsored in part by the NASA/GSFC Earth Science Data Systems Working Group.

found reusable components. The reusability section asked if respondents made any software components available for reuse by others, then followed up with questions including what factors prevented them from making components available for reuse, the types of components made available for reuse, and how often they believe the components are reused by others. The community needs section included questions on what factors would help increase the level of software reuse within the Earth science community, what artifacts they would reuse if made available, and allowed space for additional comments and questions.

The 2004 survey was sent to members of the Software Reuse Working Group and other government employees; 25 responses were received. Office of Management and Budget (OMB) approval was obtained for the survey on Jan. 4, 2005 (Approval Number 2700-0117), and the survey was distributed to a wider audience, including members of academia; 100 responses were received from about 3000 invitations to participate (approximately a 3.3% return rate).

III. SURVEY RESULTS

The results of the 2005 survey confirm the results of the 2004 survey. There were some shifts in the answers to some questions, but these were typically small. The general results are the same in both surveys, and the same conclusions can be drawn from either one. Therefore, we will focus our discussion on the 2005 survey which is more recent and received a larger number of responses. The majority of the questions in the survey asked the respondents to rate various answer choices on a 1-5 scale representing the importance or frequency of the choice, as appropriate for the question. The importance scale was typically: (1) not important at all, (2) not very important, (3) somewhat important, (4) important, and (5) very important. The frequency scale was typically: (1) never, (2) rarely, (3) sometimes, (4) often, and (5) very often. Average values were calculated for each choice and used to rank them within each question. These are used as a measure of the overall, general opinion of the survey respondents.

A. Components Reused vs. Components Made Available

One of the most interesting results of the survey regards the types of software and software development artifacts that respondents reused and how they developed the ones they made available for reuse by others. We asked six questions relating to this subject in order to determine if current practices may present a barrier to reuse. In particular, if respondents desire to use a certain type of component, but tend to develop and make available a different type of component, this would point towards a barrier that needs to be broken down in order to increase the systematic reuse of software components.

There were three questions related to reuse of existing software and software development artifacts and three related to developing software and artifacts for easier reuse by others. The first in each set simply asked if respondents had reused artifacts outside of their project/group or made artifacts available outside of their project/group within the last five years. There were 79% of the respondents who answered “yes” in the case of reusing existing software and 74% in the case of

making artifacts available for reuse. The following two questions in each set were asked only of the people who answered “yes” to the first question. These questions regarded the frequency with which certain types of artifacts or software were reused or developed for reuse. One question asked about software development artifacts: algorithms and techniques, designs and architectures, source code and scripts, executables and binaries, and other. The other question asked about types of software: complete systems or applications, subsystems or components, code libraries, code fragments, and other.

In terms of what was reused, there was a clear preference for the smaller-sized components (see Fig. 1). Algorithms, techniques, source code, and scripts were reused more often than designs, architectures, executables, and binaries. Code libraries and code fragments were reused more often than subsystems or components and complete systems or applications. A difference appeared in what was developed for reuse. For software development artifacts, the results similarly favored smaller-sized components (source code, scripts, algorithms, and techniques). However, for the types of software made available for reuse, larger-sized components were offered – subsystems or components and complete systems or applications were more frequently made available than code fragments or code libraries.

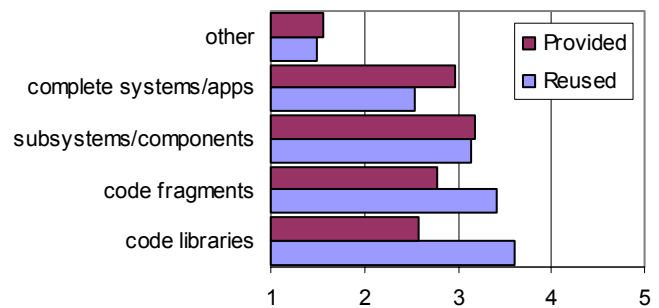


Figure 1. Types of Software Provided for Reuse and Reused

These results point to a potential problem and thus a barrier to reuse. Complete systems or applications and subsystems or components are being made available for reuse, but code libraries and code fragments are most desired for reuse purposes. The fact that there is a tendency to provide larger-sized components when smaller-sized ones are desired suggests that it will be difficult for software developers to find the types of software they want. If they are unable to find what they are looking for, they will not be able to reuse existing components, and may end up rewriting components when it is not necessary. Increasing the amount of smaller-sized components made available for reuse by others should encourage and increase the amount of reuse.

B. Reasons for not Reusing or Making Available Components

We were also interested in knowing the reasons why people did not reuse existing software or make their software available for reuse by others. We asked four questions related to this topic in order to determine what barriers to reuse existed in the experience of our survey respondents. In order to increase the amount of reuse, it is important to know what factors are currently preventing people from practicing reuse.

Two questions are the same as ones described in the previous section – the ones asking if respondents did reuse artifacts or make artifacts available for reuse by others. The other two questions were asked of the people who answered “no” to those questions (21% did not reuse software from outside of their project/group and 26% did not make artifacts available for reuse outside of their project/group). Ten choices were provided as possible reasons for not reusing existing artifacts. Eight choices were provided as possible reasons for not making artifacts available for reuse. Each question also offered an “other” option to account for reasons not listed explicitly. In addition, the same question about reasons for not making artifacts available was also asked of the 74% of the respondents that did make some artifacts available. This was to determine if the two groups of respondents faced similar barriers. One difference to note between the 2004 and 2005 surveys is that the question regarding reasons for not reusing existing software received significantly more responses in the 2005 survey; the results from the 2004 survey are too limited to provide any useful information.

The primary reasons respondents did not reuse software from outside their project/group were that they did not know where to look for reusable artifacts and they did not know suitable reusable artifacts existed at the time. The reasons respondents who did not make any software development artifacts available for reuse outside of their project/group tended to be more varied. The main reasons included not knowing if it would be useful, support and maintenance concerns, the cost of developing for reuse, no standard method for distribution, and not knowing how. Among the respondents who did make at least some artifacts available for reuse, the reasons for not making artifacts available were also varied, but were very similar. The main reasons here were support and maintenance concerns, the cost of developing for reuse, no standard method for distribution, limitations in the organization’s software release policy, and not knowing if it will be useful.

These are important results because they indicate where additional work needs to be done in order to increase the level of reuse within the Earth science community. The main barrier to reusing existing artifacts is lack of knowledge about what suitable reusable artifacts exist and where to find them. Therefore, reusable artifacts need to be more readily available and more easily locatable. If software developers know where to look for artifacts and can easily find suitable artifacts, they will be more likely to reuse them. They can then help others reuse existing artifacts by passing on their knowledge about the location and availability of such products. In a separate question to respondents who did reuse artifacts, personal knowledge from past projects and word-of-mouth or networking were the primary ways of locating and acquiring software development artifacts. (Web searches were of average importance while serendipity and reuse catalogs or repositories were rated the lowest.) The larger variety of reasons provided for not making artifacts available for reuse by others makes it more difficult to determine how to break down the barriers here. However, it also points to a variety of areas where improvements can be made.

C. *Modifying Artifacts and Licenses Used*

Another pair of questions asked only to the respondents who indicated that they had reused software development artifacts dealt with modifications to the artifacts and the licenses under which the artifacts were reused. The first question asked about (a) the frequency with which artifacts were modified and (b) the frequency with which those changes were communicated back to the original developer(s) of the artifact. The second question asked respondents to indicate the frequency with which the following licensing methods or agreements were used: open source, shareware or public domain, formal license agreement with the developer, semi-formal license agreement with the developer, no formal license agreement.

The results showed that artifacts were modified with moderate/average frequency (coded as “sometimes” on the scale used in the survey). However, the changes were communicated back to the developers with a somewhat lower frequency. In terms of licensing methods, open source software was clearly preferred over the other options. Use of shareware or public domain also rated somewhat above average frequency. All of the licensing options (formal, semi-formal, none) rated below average frequency.

The average frequency for modification of artifacts suggests that there is a relatively equal mix of artifacts that can be reused as-is, without modification, and ones that require some degree of modification to meet the requirements of a new environment. We did not ask about the amount of modification done though, so we do not have a measure of whether it is typically high or low. The preference for open source software is expected given the open nature of these licenses, which typically allow free redistribution of the software, provide access to source code, and allow modifications and derived works [4]. An interesting point was that formal license, semi-formal license, and no formal license all rated below average frequency. Theoretically, these should cover the range of possibilities in a mutually exclusive way; e.g., if you are not using a formal license or a semi-formal license, you must be using no formal license. Thus we expected to see one of these choices rate above average, one near average, and one below average. Since this is not the case, perhaps respondents did not view these three options in the same way as we did when creating them.

D. *Factors to Increase Reuse*

The final section of the survey included a few questions for all respondents. One of these was how important different specified factors would be in helping increase the level of reuse within the Earth science community. Clearly, this is an important piece of information as it provides more explicit information on where work can be done to increase reuse.

Six factors were provided, plus one “other” choice to account for any not listed. The top three factors were having an Earth science catalog/repository for reusable artifacts, use of open source licensing, and education and guidance on reuse. The three lower ranking factors were a standardized support policy for reused software, changes to NASA external release

policy, and a standardized license agreement for the Earth science community.

These results provide direction for future work, and are consistent with the opinions of the respondents as expressed in the answers to other questions. The primary reasons respondents did not reuse artifacts were that they did not know where to look or that suitable reusable artifacts existed at the time. Having a catalog/repository dedicated to Earth science software would give them a place to look for reusable artifacts while also increasing their knowledge about the existence of currently available reusable artifacts. Also, the use of reuse catalogs and repositories was rated the most important method of increasing the level of reuse within the community. This suggests that respondents would use such an Earth science catalog/repository for reusable artifacts if it existed, indicating that it would be a worthwhile endeavor to create such a system. Since a system like this would eliminate the two main reasons people did not reuse artifacts (not knowing where to look and not knowing suitable reusable artifacts existed), this is an understandable result. It also touches on another reason respondents did not make software available for reuse – no standard method of distribution. An Earth science catalog or repository for reusable artifacts could serve as a standard way of making software available to others in the community.

Open source software is already the primary choice of licensing for most respondents, so it is logical that they would also recommend greater use of open source licensing as a way to increase reuse. The general freedom of open source licensing as compared to more closed licensing mechanisms makes it an attractive and useful option for software developers. Encouraging greater use of open source licensing is another area of work where successful results can produce greater levels of reuse.

More education and guidance on reuse will also help break down existing barriers. For example, making people aware that smaller components are most desired for reuse, but larger components are more frequently made available, may help create a shift in the type of software that developers make available for reuse. It can also help people understand the usefulness of reuse, taking away a possible reason people would not make their artifacts available. One question asked respondents who did reuse artifacts about their reasons for doing so – saving time and ensuring reliability were the primary reasons, with saving money not far behind. Education can make more people aware of these points and the benefits they gain by practicing reuse. Guidance on reuse can help address issues where people do not know how to make artifacts available. Education and guidance should naturally lead to improvements as more people understand the benefits of reuse and work to break down the existing barriers to reuse.

IV. CONCLUSIONS

Our surveys provide support for the common view that software reuse saves time, saves money, and ensures the reliability of the product. These were the top three reasons respondents chose to reuse existing software. We also discovered some potential barriers to reuse. Smaller-sized

components such as code libraries and code fragments are the most desired for reuse purposes, but larger-sized components such as complete applications and subsystems are more frequently made available. Another barrier is that people did not know that suitable reusable artifacts existed or where to locate reusable artifacts. In order to help increase the amount of reuse, these barriers need to be broken down.

The responses we received also provided some information on how to increase the level of reuse. The top three suggestions were having an Earth science catalog/repository for reusable artifacts, greater use of open source licensing, and more education and guidance on reuse. Even though personal knowledge and word-of-mouth networking were the most commonly used methods of locating reusable artifacts and catalogs/repositories were the least often used methods, an Earth science catalog/repository of reusable assets was seen to be the most important method of increasing reuse. This suggests that people will use such a catalog/repository, if a suitable one existed. Such a system would also help break down the barriers of not knowing what reusable artifacts existed or where to look for them. The recommendation for greater use of open source licensing is reasonable considering the benefits it provides and that it is already the most commonly used licensing method with the Earth science community. Education and guidance on reuse should also help break down barriers by teaching people about the benefits of reuse, how to reuse existing components and how to make their own components available for reuse by others, and what barriers to reuse need to be broken down.

Software reuse is being practiced within the Earth science community, but there are still barriers to reuse and progress to be made. Further education and guidance should encourage a greater number of people to participate in reuse and help break down existing barriers. Removing barriers would further increase the level of reuse, and help make systematic software reuse a more regular and routine part of the software development process.

ACKNOWLEDGMENT

We would like to thank all of the survey respondents for taking the time to complete our survey and providing us with valuable information about reuse practices within the Earth science community. Also, thanks are given to the reuse working group members and support team for their help in developing the survey and analyzing survey results.

REFERENCES

- [1] R. E. Grinter, "From local to global coordination: lessons from software reuse," *Proceedings of the 2001 International ACM SIGGROUP Conference on Supporting Group Work*, pp. 144-153, 2001.
- [2] J. Long, "Software Reuse Antipatterns", *ACM SIGSOFT Software Engineering Notes*, vol. 26, iss. 4, pp. 68-76, July 2001.
- [3] S. Samadi, N. Alameh, R. Wolfe, S. Olding, D. Isaac, "Strategies for Enabling Software Reuse within the Earth Science Community", *Proceedings of the 2004 IEEE International Geoscience and Remote Sensing Symposium*, vol. 3, pp. 2196-2199, September 2004.
- [4] B. Perens, "The Open Source Definition Version 1.9", <http://www.opensource.org/docs/definition.php>, accessed Apr. 21, 2006.